Mouse
#6: MouseText Characters

Revised by: Matt Deatherage January 1989
Revised by: Rilla Reynolds November 1985

This Technical Note describes the MouseText character set which is available on all currently produced Apple II computers.
Changed since November 1988: Corrected typographical errors in the BASIC and assembly language program examples.

In unenhanced Apple IIe computers, the alternate character set contained two sets of inverse uppercase characters. In the enhanced Apple IIe, and in all Apple IIc and IIGS computers, one set of inverse uppercase characters is replaced by a MouseText character set. MouseText is a set of graphical characters designed to allow Apple II computers to display a desktop metaphor on the text screen. The Apple II Desktop Toolkit uses these characters, as do applications like AppleLink–Personal Edition.

If your program used the set of inverse uppercase characters which were replaced by MouseText (the set mapped to ASCII values $40-$5F), your program will display MouseText characters instead of inverse uppercase characters on all currently-produced Apple II computers. If your program used the other set of inverse uppercase characters (ASCII values $00-$1F), it will display inverse capital characters as expected.

The following will help you identify if the changes affect you or not.

1. If your program is written entirely in BASIC or Pascal or your assembly language program calls the `COUT` routine to put characters on the screen, you are not affected. The only exception would be if you print (`POKE`) inverse characters directly to the text screen in BASIC.
2. If your program uses the standard character set (checkerboard cursor) you are not affected.
3. If your program is using the alternate character set (solid cursor) and is directly storing values (via `POKE`) to the text display area, you will encounter problems if your character values are in the range from 64 ($40) to 95 ($5F). To recreate the original display, use values in the range from 0 ($0) to 31 ($1F) instead. Note that these lower values display as inverse uppercase characters on older machines as well.

Following are the methods recommended for accessing MouseText characters from various languages:

AppleSoft BASIC

1. Turn on the video firmware with `PR#3` (if under DOS 3.3 or ProDOS, use `PRINT CHR$(4);"PR#3"`)
2. Enable MouseText characters by printing an ASCII 27 ($1B) to the screen.
3. Set inverse printing mode by printing an ASCII 15 ($0F) to the screen.

To stop displaying MouseText characters:

1. Disable MouseText characters by printing an ASCII 24 ($18) to the screen.
2. Set normal print mode (if desired) by printing an ASCII 14 ($0E) to the screen.

This short BASIC program displays all MouseText characters under DOS 3.3 and ProDOS:

```
10     D$=CHR$(4)
20     PRINT D$;"PR#3": REM Turn on the video firmware
30     PRINT:REM This is so the screen won't be in inverse
40     PRINT CHR$(15):REM Set inverse mode
50     PRINT CHR$(27);"ABCEDFGHIJKLMNOPQRSTUVWXYZ@[]^_\";CHR$(24)
60     PRINT CHR$(14):END
```

Assembly Language

Assembly language programs are expected to follow the same procedure as AppleSoft BASIC. Use calls to `COUT` to print MouseText characters to the screen. The following is a sample assembly language program which displays two MouseText characters (which create a folder icon), along with their inverse uppercase equivalents:

```
START               LDA #$A0            ;USE A BLANK SPACE TO
                    JSR $C300           ;TURN ON THE VIDEO FIRMWARE
                    LDY #0              ;INITIALIZE COUNTER
LOOP                LDA STR,Y           ;GET VALUE
                    JSR $FDED           ;SEND IT THROUGH THE COUT ROUTINE
                    INY
                    CPY STRLEN
                    BNE LOOP            ;=>NOT DONE YET
                    RTS
STR                 DFB $1B,$58,$59,$18,$58,$59
                                        ;MOUSETEXT ON, SHOW, MOUSETEXT OFF, SHOW
STRLEN              EQU *-STR           ;LENTGH OF STR
```

Note: Using MouseText on the text screen by directly poking or storing MouseText character values into the text buffer is not supported by Apple at this time. Should the MouseText character set require remapping in the future, those programs which use the methods outlined in this Note should still work with any new mapping. Those which directly store MouseText values run the strong risk of display failure under a new mapping.

Apple II Pascal

1. Output a CHR(27), an escape character, to enable MouseText.
2. Output a CHR(15) to turn on inverse video.
3. Output the appropriate capital letter for the desired MouseText character.

A Pascal sample program:

```
PROGRAM OUTPUT_MOUSETEXT
VAR CMD:PACKED ARRAY[0..1] OF 0..255
BEGIN
       CMD[0]:=27; CMD[1]:=15;
       UNITWRITE(1,CMD,2); {turn on MouseText mode}
       {code to display MouseText
                            .
                            .
                            .
       }
       CMD[0]:=24;
       UNITWRITE(1,CMD,1); {turn off MouseText mode}
END
```

Pictorial descriptions of the MouseText character set may be found in the *Apple IIe Technical Reference Manual*, the *Apple IIc Technical Reference Manual, Second Edition*, and the *Apple IIGS Hardware Reference*.

Note: The pictures of MouseText characters in these manuals differ from early implementations. In early MouseText character sets, the icons mapped to the letters F and G combined to form a "running man." In current production, these letters are different pictures (an inverse carriage return symbol and a window title bar pattern) which form no picture when placed next to each other. Programs should not attempt to use the running man MouseText characters.

Further Reference

- *Apple IIGS Hardware Reference*
- *Apple IIe Technical Reference Manual*
- *Apple IIc Technical Reference Manual, Second Edition*